

## Full-order optimal compensators for flow control: the multi-input case.

Onofrio Semeraro · Jan O. Pralits

the date of receipt and acceptance should be inserted later

**Abstract** Flow control has been the subject of numerous experimental and theoretical works. In this numerical study, we analyse full-order, optimal controllers for large dynamical systems in presence of multiple actuators and sensors. We start from the original technique proposed by [1], the *adjoint of the direct-adjoint* (ADA) algorithm. The algorithm is iterative and allows bypassing the solution of the algebraic Riccati equation associated with the optimal control problems, typically unfeasible for large systems.

We extend ADA into a more generalized framework that includes the design of multi-input, coupled controllers and robust controllers based on the  $\mathcal{H}_\infty$  framework. The full-order controllers do not require any preliminary step of model reduction or low-order approximation: this feature allows to pre-assess the optimal performances of an actuated flow without relying on any estimation process or further hypothesis.

We show that the algorithm outperforms analogous technique, in terms of convergence performances considering two numerical cases: a distributed system and the linearized Kuramoto-Sivashinsky equation, mimicking a full three-dimensional control setup. For the ADA algorithm we find excellent scalability with the number of inputs (actuators) in terms of convergence to the solution, making the method a viable way for full-order controller design in complex settings.

---

Onofrio Semeraro  
Department of Mechanics, Mathematics and Management (DMMM), Politecnico di Bari,  
Bari, Italy (E-mail: onofriosem@gmail.com)

Jan O. Pralits  
Department of Chemical, Civil and Environmental Engineering (DICCA), University of  
Genoa, Genoa, Italy (E-mail: jan.pralits@unige.it)

## 1 Introduction

Flow control based on linear strategies has been applied to a large variety of flows in the last decades, see [2], [3], [4], [5]. The interest in such applications has been invigorated by the possible, numerous outcomes, ranging from drag reduction to acoustic emission mitigation. From a physical point of view, one of the main assumptions is that the modification of the coherent structures can be achieved by properly acting on the flow, both in the presence of small amplitude perturbations or large coherent structures at higher Reynolds number. If a linear approximation is valid, rigorous methods for linear control can be applied.

In this paper we focus on active control. By definition active control is characterized by energy input in the system to be controlled, by means of actuators. The system to be controlled will often be referred to as *plant*. The presence of actuators allows more flexibility in the control design: sensors can be introduced such that the actuation is optimized with respect to an objective function.

An example of a plant to be controlled using active control strategies is given in Fig 1, where the sketch of a boundary layer flow developing on a flat plate is shown, including  $m$  actuators  $\mathbf{B}_u$  and  $p$  sensors  $\mathbf{C}_z$ . We seek for a control law  $\mathbf{u}(t)$ , feeding the  $m$  actuators. A possible way, it is to define a *control kernel*  $\mathbf{K} \in \mathbb{R}^{m \times n}$  such that the control signal  $\mathbf{u}(t)$  – defined in a time interval  $t = [0, T]$  – is proportional to the state vector  $\mathbf{q} \in \mathbb{R}^n$ , *i.e.*  $\mathbf{u}(t) = \mathbf{K}\mathbf{q}$ . The controller is designed for fulfilling a target; in linear quadratic regulators (LQR) the following objective function to be minimized can be defined as follow

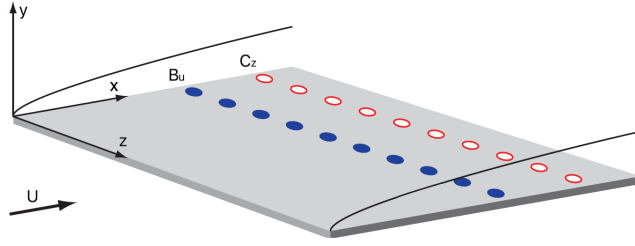
$$\mathcal{J} = \frac{1}{2} \int_0^T (\mathbf{z}^H \mathbf{z} + \mathbf{u}^H \mathbf{R} \mathbf{u}) dt. \quad (1)$$

The first term includes the signals recorded at the location of the sensors  $\mathbf{C}_z$ , as  $\mathbf{z} = \mathbf{C}_z \mathbf{q}$ . The matrix  $\mathbf{R}$  contains the control penalties for tuning the control effort (see for instance [6], [7] and reference therein).

A difficulty that arises in fluid mechanics is related to the dimensions of the dynamical systems: although it is relatively common to deal with numerical simulations with a number of degrees of freedom  $n > 10^6$ , control design tools become infeasible for much smaller dimensions. A classical approach to circumvent this limitation consists in replacing the full-order system with reduced-order models capturing the *essential* dynamics of the system. This methodology is sometimes called *reduce-then-design*.

### 1.1 Full-order control design

Despite the necessity of reduced-order modelling for real application, the design of full-order controllers can be of great interest. Indeed, the reduced-order modelling is often performed in *open-loop*, thus discarding states of the system



**Fig. 1** Sketch of active control for boundary layer flows; the blue dots indicate a row of localized actuators ( $B_u$ ), the red circles are the sensors ( $C_z$ ) for the performance evaluation. The aim of the actuators is to modify the flow in order to fulfil a control objective. Adapted from [4]

that might be reachable in the presence of a controller; the un-modelled states can lead to inconsistencies in the control design. Moreover, when designing a controller based on reduced-order models, an estimator is often necessary: this leads to a quite larger parameters space to be explored, that includes numerous choices such as the location and the spatial distribution of the sensors/actuators pairs. In this sense, it is not always straightforward to assess whether a system is controllable and – when possible – if the achieved performances are the best possible due to these numerous choices that must be done during the control design. For this reason, we believe that it is of interest to pre-assess the performance of a controller by minimizing the number of these choices using optimal full-order controllers that do not require any preliminary step of model reduction or estimation processes.

Within the optimal control framework, the control gain  $K$  can be obtained as solution of the algebraic Riccati equation (ARE), [8]; however, already for systems of dimensions  $n \approx 10^3 - 10^4$  the direct method is computationally intractable [9]. Neglecting the low-order design, alternative methods for the full-order design were proposed in the last decades, based on the solution of surrogate systems of equations, iterative procedures or algorithms exploiting the inherent sparsity of the considered systems. A common trait of these techniques is their feasibility when the number of actuators  $m \ll n$ . A classic technique is the Chandrasekhar method: the full-order ARE is replaced by a set of partial differential equations. Alternatively, the projection on low-order Krylov spaces by means of Arnoldi-type processes (see [10] and references therein) and Newton methods have been proposed ([11]). An efficient solution method for this set of equations was proposed in [12], where long-time integrators are used in combination with reduced-order models based on proper orthogonal decomposition (POD) for the control of the Karman street developing behind a cylinder. Full-order kernels were computed for the control of the plane channel by [13], based on the works by [14]; in this method, optimal control kernels with a compact support in physical space are designed

in the limit of parallel flow, by solving an optimal control problem for each wavenumber pair in Fourier space, independently. A Fourier anti-transform allows reconstructing the kernel in physical domain. The procedure was applied to weakly spatially developing flows by [15] and [16], although the main drawback is the distribution of sensors/actuators pair that cannot be localized in this approach. Methods based on the pole placement are discussed, as the minimum-energy control (MCE), already introduced in [1] and [17]; in this limit, the control kernel is defined starting from the unstable adjoint model of the system. Examples are given by [18] for the control of vortex shedding behind a cylinder, and [19] for the control of the instabilities rising in the wake of a thick plate at higher Reynolds number. In the latter contribution, the linearization is performed around different mean flows, computed as RANS and U-RANS solutions.

## 1.2 Iterative methods and present investigation

Finally, iterative methods can be considered. The standard approach consists of an iteration based on the direct equation describing the system dynamics and its adjoint (see [20] and citations therein). The aim of this technique is the identification of the optimal control law  $u(t)$ , and it can be generalized to nonlinear settings as well as receding horizon control problem ([21], [22], [23]).

However, the control signal in the standard direct-adjoint iteration (hereafter indicated with DA) depends on the initial condition from which the trajectory emanates. This limitation is tackled in Bewley *et al.* [1], where the *adjoint of the direct-adjoint* (ADA) algorithm is introduced, a method capable of identifying the kernel gain  $\mathbf{K}$ . Examples of this application can be found for the optimal control of the von Kármán street developing past a cylinder in [24] or the control of Tollmien-Schlichting waves developing in a two-dimensional boundary layer flow in [25].

From the theoretical point of view, the ADA algorithm replaces the original direct-adjoint optimization with the corresponding sensitivity analysis by considering the adjoint of the entire problem: this procedure changes an optimization problem of size  $n$ , the number of states, into a problem of size  $m$ , the number of inputs. The algorithm is detailed in Sec. 2, where the original formulation is revisited. In this article we focus on two aspects of the algorithm. First, we thoroughly reconsider the design of a multi-input controller based on ADA; in Sec. 3, we extend the algorithm to the multi-input case, by considering the *centralized* control design [26]. This allows also the application of the same framework to the robust optimal control,  $\mathcal{H}_\infty$  [27]. Moreover, we assess the performance of the algorithm from the convergence point of view; we compare the multi-input ADA algorithm is compared to an analogous iterative methodology introduced in [28], based on stochastic gradient (Sec. 4.3). The comparison makes use of a simple distributed system. In Sec. 5, we apply the multi-input ADA to a two-dimensional version of the Kuramoto-Sivashinsky (KS) equation. The work finalizes with conclusions in Sec. 6.

## 2 Linear Quadratic Regulator (LQR): iterative solution and Riccati equation

In this section, we concisely state the optimal control problem and derive the time-continuous algebraic Riccati equation (CARE). For a deeper discussion, we refer to the specialized literature [8]. We introduce the input-output equations

$$\frac{d\mathbf{q}}{dt} = \mathbf{A}\mathbf{q} + \mathbf{B}_u\mathbf{u}, \quad \text{with} \quad \mathbf{q}(0) = \mathbf{q}_0, \quad (2a)$$

$$\mathbf{z} = \mathbf{C}_z\mathbf{q}, \quad (2b)$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is the system matrix of dimensions  $n$ , the degrees of freedom. In this work, we consider time-continuous, spatially-discretized systems. The spatial distribution of the actuators is described by the matrix  $\mathbf{B}_u \in \mathbb{R}^{n \times m}$ . The variable  $\mathbf{z}$  is scalar and represent a time signal. If the system arises from the discretization of the Navier-Stokes equations linearized around a baseflow or a mean flow, the vector  $\mathbf{q}(t) \in \mathbb{R}^n$  represents the state of the fluid system. We want to identify a control signal  $\mathbf{u}(t)$  such that

$$\mathcal{J} = \frac{1}{2} \int_0^T (\mathbf{q}^H \mathbf{Q} \mathbf{q} + \mathbf{u}^H \mathbf{R} \mathbf{u}) dt + \frac{1}{2} \mathbf{q}(T)^H \mathbf{Q}_T \mathbf{q}(T) \quad (3)$$

is minimized. The matrices  $\mathbf{Q} \geq 0 \in \mathbb{R}^{n \times n}$ ,  $\mathbf{R} > 0 \in \mathbb{R}^{m \times m}$  and  $\mathbf{Q}_T \geq 0 \in \mathbb{R}^{n \times n}$  contain weights in the entries. The matrix  $\mathbf{Q}$  can be low-rank, for instance by defining it based on the sensors  $\mathbf{C}_z$  (see Fig. 1 and Eq. 1). In the following, the matrices  $\mathbf{Q}$  and  $\mathbf{R}$  are assumed diagonal, and the final condition  $\mathbf{Q}_T$  is set null. These assumptions do not lead to any loss of generality.

The solution of the control problem in Eq. 2-3 can be obtained by defining an augmented cost function  $\tilde{\mathcal{J}}$

$$\tilde{\mathcal{J}} = \mathcal{J} - \int_0^T \mathbf{p}^H \left( \frac{d\mathbf{q}}{dt} - \mathbf{A}\mathbf{q} - \mathbf{B}_u\mathbf{u} \right) dt. \quad (4)$$

Applying integration by parts, the following system of equations is cast

$$\frac{d\mathbf{q}}{dt} = \mathbf{A}\mathbf{q} + \mathbf{B}_u\mathbf{u}, \quad \text{with} \quad \mathbf{q}(0) = \mathbf{q}_0, \quad (5a)$$

$$\frac{d\mathbf{p}}{dt} = -\mathbf{A}^H \mathbf{p} - \mathbf{Q}\mathbf{q}, \quad \text{with} \quad \mathbf{p}(T) = 0, \quad (5b)$$

$$\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{u}} = \mathbf{B}_u^H \mathbf{p} + \mathbf{R}\mathbf{u}. \quad (5c)$$

The equation for the adjoint state  $\mathbf{p}(t) \in \mathbb{R}^n$  is obtained by zeroing the gradient  $\partial \tilde{\mathcal{J}} / \partial \mathbf{q}$  and is integrated backward in the interval  $t \in [T, 0]$ . The matrix  $\mathbf{A}^H$  denotes the adjoint operator, satisfying the inner-product  $\langle \mathbf{A}\mathbf{q}, \mathbf{p} \rangle = \langle \mathbf{q}, \mathbf{A}^H \mathbf{p} \rangle$ . The unknown of the system is the control signal  $\mathbf{u}(t)$  in the time interval

$t \in [0, T]$ . The solution can be approximated by iteration: at each step of the direct-adjoint iteration the control signal is updated as

$$\mathbf{u}(t)^{i+1} = \mathbf{u}(t)^i - \beta^i \left( \frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{u}} \right)^i, \quad (6)$$

using 5(c). The resulting control signal is optimal for a given initial condition  $\mathbf{q}_0$ . A gradient descent algorithm can be used for defining the step  $\beta$  [29]. We will refer to this technique as direct-adjoint (DA) iteration in what follows.

A direct solution of the control problem is obtained by solving the associated Riccati equation. The system in Eq. 5 can be written in matrix form as

$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{pmatrix} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}_u \mathbf{R}^{-1} \mathbf{B}_u^H \\ -\mathbf{Q} & -\mathbf{A}^H \end{bmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix}.$$

Assuming the relation  $\mathbf{p} = \mathbf{X}\mathbf{q}$ , the following Riccati equation is obtained

$$\mathbf{A}^H \mathbf{X} + \mathbf{X} \mathbf{A} - \mathbf{X} \mathbf{B}_u \mathbf{R}^{-1} \mathbf{B}_u^H \mathbf{X} + \mathbf{Q} = 0, \quad (7)$$

for linear time-invariant system, in the steady case limit  $\dot{\mathbf{X}} = 0$ . The matrix  $\mathbf{X} \in \mathbb{R}^{n \times n}$  is the solution of the Riccati equation and it is positive-definite and symmetric. The control signal  $\mathbf{u}(t)$  is proportional to the state  $\mathbf{q}$  as  $\mathbf{u}(t) = \mathbf{K}\mathbf{q}$ ; the constant control kernel  $\mathbf{K} \in \mathbb{R}^{m \times n}$  is

$$\mathbf{K} = -\mathbf{R}^{-1} \mathbf{B}_u^H \mathbf{X}. \quad (8)$$

For large-system  $n > 10^3$  the direct solution of the Riccati equation is not feasible, due to computational costs of order  $O(n^3)$  – regardless of the structure of the system matrix  $\mathbf{A}$  – and storage requirements which are at least of order  $O(n^2)$ , see [10]. A viable alternative is represented by iterative methods for the computation of the control kernel  $\mathbf{K}$ .

## 2.1 Adjoint of the Direct-Adjoint (ADA) algorithm for the solution of the LQR problem

The DA iteration does not allow the direct computation of the optimal control kernel  $\mathbf{K} \in \mathbb{R}^n$ ; indeed, the unknowns of the problem for a given initial condition  $\mathbf{q}_0^i$  is the control signal  $\mathbf{u}(t)^i$  defined in  $t \in [0, T]$ , with  $T$  the final time of optimization. However, as observed by [1], the following linear system can be formed

$$[\mathbf{u}_0^1 \ \mathbf{u}_0^2 \ \dots \ \mathbf{u}_0^n] = \mathbf{K}_{(1 \times n)}, [\mathbf{q}_0^1 \ \mathbf{q}_0^2 \ \dots \ \mathbf{q}_0^n]_{(n \times n)}, \quad (9)$$

where  $n$  different solutions  $\mathbf{u}(t)$  of the optimal control problem emanating from  $n$  linearly independent initial conditions  $\mathbf{q}_0^i$  are used. Thus,  $n$  iterations need to be solved, each with a different initial condition. In particular, the

known vector on the left-hand side can be formed by taking the corresponding control signal at  $t = 0$ . The columns of the matrix on the right-hand side are represented by the initial conditions of each of the  $n$  iterative loops. The last step consists of the solution of a linear system of dimensions  $n \times n$ ; so, for large system this iterative scheme is again unfeasible.

However, as elucidated in [24] and [1], one may drastically reduce the computational costs of the problem by analysing the sensitivity with respect to the initial condition using the *adjoint* of the DA system. The original demonstration of the algorithm makes use of integration by-parts and it is reported in the appendix A for sake of completeness. Here, we propose an alternative version based on the properties of the Hamiltonian systems. Introducing the symplectic matrix

$$\mathbf{J} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix},$$

the following property is fulfilled

$$\mathbf{T}^H \mathbf{J} + \mathbf{J} \mathbf{T} = \mathbf{0}. \quad (10)$$

The matrix  $\mathbf{T} \in \mathbb{R}^{2n \times 2n}$  for the control problem in Eq 2 is defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{A} & -\mathbf{B}_u \mathbf{R}^{-1} \mathbf{B}_u^H \\ -\mathbf{Q} & -\mathbf{A}^H \end{bmatrix}.$$

Let us introduce the state  $\mathbf{x} = \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix}$  as solution of the system  $\dot{\mathbf{x}} = \mathbf{T} \mathbf{x}$ , with initial condition  $\mathbf{x}_0 = \begin{pmatrix} \mathbf{q}_0 \\ \mathbf{p}_T \end{pmatrix}$  and its *adjoint* state  $\tilde{\mathbf{x}} = \begin{pmatrix} \tilde{\mathbf{q}} \\ \tilde{\mathbf{p}} \end{pmatrix}$ , solution of the system  $\dot{\tilde{\mathbf{x}}} = \mathbf{T}^H \tilde{\mathbf{x}}$ , with initial condition  $\tilde{\mathbf{x}}_0 = \begin{pmatrix} \tilde{\mathbf{q}}_0 \\ \tilde{\mathbf{p}}_T \end{pmatrix}$ . Note that the state  $\tilde{\mathbf{x}}$  is the adjoint of the direct state  $\mathbf{x}$  with respect to the symplectic product  $\Omega(t)$ , defined as

$$\Omega(t) = \tilde{\mathbf{x}}(t)^H \mathbf{J} \mathbf{x}(t) = \tilde{\mathbf{p}}(t)^H \mathbf{q}(t) - \tilde{\mathbf{q}}(t)^H \mathbf{p}(t). \quad (11)$$

In Hamiltonian systems, this product is constant  $\forall t$ ; including the boundary conditions, we obtain  $\Omega(t) = 0$ . The relation 11 can be compared with the optimality condition at  $t = 0$

$$\tilde{\mathbf{p}}_0^H \mathbf{q}_0 = \tilde{\mathbf{q}}_0^H \mathbf{p}_0, \quad (12a)$$

$$\mathbf{u} = \mathbf{K} \mathbf{q}_0 = (-\mathbf{R}^{-1} \mathbf{B}_u^H) \mathbf{p}_0. \quad (12b)$$

We can notice that, if we introduce as initial condition of the dual system  $\tilde{\mathbf{q}}_0$  one row of  $-\mathbf{R}^{-1} \mathbf{B}_u^H$ , the resulting adjoint solution  $\tilde{\mathbf{p}}_0^H$  corresponds to one row of  $\mathbf{K}$ . Thus, the solution of the Riccati problem is obtained without solving the algebraic equation Eq. 7. In principle, the exact solution is obtained for  $T \rightarrow \infty$ . In practice, a sufficiently long time-window for the optimization guarantees convergence towards the optimal solution. Strictly speaking, the resulting problem is not anymore an optimal control problem, despite it makes use of the same equations.

### 3 The multi-input version of ADA

In this section we discuss how to extend the ADA algorithm to multivariable systems, *i.e.* the systems characterized by the presence of multiple inputs and outputs. Within the context of flow control, this is typically the case of full, three-dimensional setups extended in the spanwise direction. An example is sketched in Fig. 1. When only one actuator/sensor pair is considered, we usually refer to a Single-Input-Single-Output (SISO) system; vice versa, the opposite case is when multiple inputs/outputs are introduced, *i.e.* the Multi-Input-Multi-Output (MIMO) case. In the forthcoming, we consider the MIMO case. The main issues for the MIMO design are introduced in Sec. 3.1. The extension of the ADA algorithm for the solution of the optimal control is discussed in section Sec. 3.2; the dual iteration for the estimation is briefly introduced in Sec. 3.3. Finally, the robust, optimal control design is revisited in Sec. 3.4, where the multivariable extension of ADA is adapted for the  $\mathcal{H}_\infty$  design ([30]).

#### 3.1 Multivariable systems

From the design point-of-view, a multivariable system is potentially characterized by cross coupling between inputs and outputs. This cross-coupling is the root of difficulties in multivariable control [26]: indeed, a change in one input can affect multiple outputs.

A first, simplistic approach for the MIMO design consists of designing a number of SISO closed-loop that equals the number of sensor/actuator pairs; considering the example in Fig. 1,  $m = p$  closed loop can be designed for each of the actuators, based on one sensor located at the same upstream location  $z_0$ . This approach is called *decentralized* approach. In this case the number of actuators must equal the number of sensors, unless neglecting some of the elements; more importantly, the cross-couplings are disregarded. The cross-couplings may of course affect both stability and performance of the closed loop system. In general, the stability of the closed-loop is not guaranteed. However, if the decentralized controller is stable in each SISO loop and weak couplings characterize the original plant, then the closed loop is also stable. In terms of performance, the resulting controller will be sub-optimal.

The opposite approach consists of accounting all the possible couplings. This methodology is called *centralized control* design. This choice guarantees optimal performances and stability of the closed loop, although it is less easy to implement in practical situations.

In the following, we discuss only these two limits. In general, linking the sensor signals with the controller that have the strongest interactions allow for hybrid solutions where a number of sensors are wired with a number of actuators; this is the so-called *pairing problem*.



### 3.2 Centralized controllers using ADA

Here, we analyse the centralized and decentralized design with respect to the optimal problem. The two cases can be summarized as follows

1. *Decentralized control*:  $m$  control gains of dimension  $n$  are designed independently from each other. Thus,  $m$  Riccati equations are solved.
2. *Centralized control*: one control gain of dimensions  $m \times n$  is designed, by solving one Riccati equation.

The application of the ADA algorithm for the decentralized case is straightforward and it is the one *de facto* discussed by [1]. Each of the  $m$  SISO Riccati equation is replaced by the corresponding ADA iteration, with one input and one output. As already mentioned, this case is only suboptimal and does not guarantee the stability of the closed loop.

The ADA algorithm does not allow identifying a low-rank approximation for the matrix  $\mathbf{X}$ ; thus, the design of the centralized controller needs to be performed row-by-row, using a number of iteration-loops that equals the number of rows. The design of a control kernel of dimensions  $\mathbf{K} \in \mathbb{R}^{m \times n}$  requires  $m$  iteration loops. The cross-coupling are accounted for by choosing the actuators and the sensors to be coupled in the input system and optimization cost function, respectively,

$$\frac{d\mathbf{q}}{dt} = \mathbf{A}\mathbf{q} + \mathbf{B}_{u,m}\mathbf{u}_m, \quad (13)$$

$$\mathcal{J} = \frac{1}{2} \int_0^T \left( \mathbf{q}^H \mathbf{C}_{z,p}^H \mathbf{C}_{z,p} \mathbf{q} + \mathbf{u}_m^H \mathbf{R} \mathbf{u}_m \right) dt. \quad (14)$$

where we indicate with  $\mathbf{B}_{u,m}$  the matrix including all the  $m$  actuators and  $\mathbf{C}_{z,p}$  all the sensors of the original plant. The initial condition of each of the iterations is chosen taking  $\tilde{\mathbf{q}}_{0i}^H = -\mathbf{R}_{i,i}^{-1} \mathbf{B}_i^H$ , such that each of the  $i$ -th rows of the matrix  $\mathbf{K}$  is obtained as adjoint solution of the iteration process.

This design strategy guarantees the coupling among the multiple inputs and corresponds to the solution of a centralized problem. In principle, following the pairing problem, one can select the pairs of sensors/actuators that share the strongest interactions *a-priori*. In the limit where only one actuator is considered in each of the  $m$  loops, we recover the decentralized case where each of the actuator is designed independently from each other.

### 3.3 Estimation problem

The technique can be extended to any problem based on the solution of algebraic Riccati equations. A first example is provided by the estimation problem as shown in [25] for the single-input-single-output setting. An estimator is defined as the following dynamical system

$$\dot{\hat{\mathbf{q}}} = \mathbf{A}\hat{\mathbf{q}} + \mathbf{B}_u \mathbf{u} - \mathbf{L}(\mathbf{y} - \hat{\mathbf{y}}), \quad (15a)$$

$$\hat{\mathbf{y}} = \mathbf{C}_y \hat{\mathbf{q}}. \quad (15b)$$

The estimator allows to reconstruct the original state  $\mathbf{q}$ , starting from local measurements  $\mathbf{y}$ ; the state  $\hat{\mathbf{q}}$  is defined such that the error  $\mathbf{y} - \hat{\mathbf{y}}$  is minimized. The estimator is driven by the error term via the matrix  $\mathbf{L}$ , referred to as estimation gain. The estimation gain  $\mathbf{L}$  is the unknown of the associated estimation problem, obtained as solution of the following Riccati equation

$$\mathbf{A}\mathbf{Y} + \mathbf{Y}\mathbf{A}^H - \mathbf{Y}\mathbf{C}_y^H \mathbf{G}^{-1} \mathbf{C}_y \mathbf{Y} + \mathbf{B}_d \mathbf{W} \mathbf{B}_d^H = 0, \quad (16a)$$

$$\mathbf{L} = -\mathbf{Y}\mathbf{C}_y^H \mathbf{G}^{-1}. \quad (16b)$$

The  $\mathbf{G}$  contains in each of the diagonal entries the estimation penalty, similarly to the control case. From the mathematical point of view, a fictitious adjoint problem is cast such that the cost function

$$\mathcal{N}(\mathbf{p}(\tilde{\mathbf{y}}), \tilde{\mathbf{y}}) = \frac{1}{2} \int_0^T (\mathbf{p}^H \mathbf{W} \mathbf{p} + \tilde{\mathbf{y}}^H \mathbf{G} \tilde{\mathbf{y}}) dt \quad (17)$$

is mimized. In this dual system, the output  $\mathbf{C}_y^H$  is an input of the system, while the adjoint of the inputs  $\mathbf{B}_d^H$  and  $\mathbf{B}_u^H$  play the role of the outputs [31]. The analogy is completed by observing that the “feedback law” is now represented by  $\tilde{\mathbf{y}}(t) = \mathbf{L}^H \mathbf{p}(t)$ . Thus, the first term of the cost function expresses the energy of the dual state  $\mathbf{p}$ , while the second term minimizes the energy input of the feedback law.

A justification of this deterministic approach is given by [3]. The physical interpretation of the method is given within the stochastic framework [32]. In this case, the unknown of the Riccati equation (16) is the expected energy of the estimation error  $\mathbf{e}(t)$ , while  $\mathbf{W}$  is the covariance of the forcing.

The optimization machinery employed for the control problem can be adopted in an analogous manner for computing the full-dimensional estimation gain  $\mathbf{L}$ . The full formulation is given in [25], where the algorithm is referred to as *Adjoint of the Adjoint-Direct* (AAD). Also the AAD algorithm can be generalized to the centralized/decentralized case. Thus, in presence of  $p$  estimation sensors  $\mathbf{C}_y$ ,  $p$  estimators can be designed based only on one sensor, independently from each other, or a centralized gain can be designed by running an iteration for each of the column of  $\mathbf{L}$ , based on all the sensors  $\mathbf{C}_y$ .

### 3.4 Riccati solution for $\mathcal{H}_\infty$ problem

The optimal control framework can be extended to the solution of robust optimization problems, introducing the *worst* disturbance scenario. The cost function associated with the problem is the following

$$\mathcal{J} = \frac{1}{2} \int_0^T \left( \mathbf{q}^H \mathbf{Q} \mathbf{q} + \mathbf{u}^H \mathbf{R} \mathbf{u} - \gamma^{-2} \mathbf{d}^H \mathbf{W} \mathbf{d} \right) dt, \quad (18)$$

where the extra-term to maximize is the unknown, worst disturbance  $\mathbf{d}$ . The optimization process consists of a simultaneous optimization problem: the signal  $\mathbf{u}$  is computed such that the a worst-case disturbance is minimized (*min-max* optimization). The parameter  $\gamma$  is chosen by the user. In this sense, the

resulting control is sub-optimal. For values of  $\gamma \rightarrow \infty$ , we approach the limit of the full-order LQR problem already considered. The direct equation of the problem reads

$$\frac{d\mathbf{q}}{dt} = \mathbf{A}\mathbf{q} + \mathbf{B}_d\mathbf{d} + \mathbf{B}_u\mathbf{u}, \quad \text{with } \mathbf{q}(0) = \mathbf{q}_0. \quad (19)$$

By introducing the corresponding augmented Lagrangian, it is possible to define the adjoint equation

$$\frac{d\mathbf{p}}{dt} = -\mathbf{A}^H\mathbf{p} - \mathbf{Q}\mathbf{q}, \quad \text{with } \mathbf{p}(T) = 0. \quad (20)$$

Two optimal conditions are defined, by zeroing the respective gradients. The first condition corresponds to the optimal control signal  $\mathbf{u}$ ,

$$\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}_u^H\mathbf{p}; \quad (21)$$

the second condition is the worst disturbance  $\mathbf{d}$

$$\mathbf{d} = \gamma^2\mathbf{W}^{-1}\mathbf{B}_d^H\mathbf{p}. \quad (22)$$

An ARE can be obtained also for this robust, optimal control problem by imposing the equivalence  $\mathbf{p} = \mathbf{X}\mathbf{q}$ , into the Hamiltonian system

$$\mathbf{T} = \begin{bmatrix} \mathbf{A} & \gamma^2\mathbf{B}_d\mathbf{W}^{-1}\mathbf{B}_d^H - \mathbf{B}_u\mathbf{R}^{-1}\mathbf{B}_u^H \\ -\mathbf{Q} & -\mathbf{A}^H \end{bmatrix}.$$

The solution is obtained by solving the following problem, based on a ARE equation

$$\mathbf{A}^H\mathbf{X} + \mathbf{X}\mathbf{A} - \mathbf{X} \left( \mathbf{B}_u\mathbf{R}^{-1}\mathbf{B}_u^H - \gamma^2\mathbf{B}_d\mathbf{W}^{-1}\mathbf{B}_d^H \right) \mathbf{X} + \mathbf{Q} = 0, \quad (23a)$$

$$\mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}_u^H\mathbf{X}, \quad (23b)$$

$$\mathbf{Y} = \gamma^2\mathbf{W}^{-1}\mathbf{B}_d^H\mathbf{X}. \quad (23c)$$

The control gain  $\mathbf{K}$  and the worst disturbance  $\mathbf{Y}$  are computed based on the solution  $\mathbf{X}$  of the ARE equation.

The ADA algorithm can be applied for the full-order approximation of this problem. Considering the single-input-single-output setting, two iterations are necessary for computing, as adjoint solution at final time  $T$ ,  $\mathbf{K}$  and  $\mathbf{Y}$  introducing as initial conditions

$$\mathbf{q}_{\mathbf{K},0} = -\mathbf{R}^{-1}\mathbf{B}_u^H, \quad (24a)$$

$$\mathbf{q}_{\mathbf{Y},0} = \gamma^2\mathbf{W}^{-1}\mathbf{B}_d^H. \quad (24b)$$

Indeed, the robust problem can be seen as a generalized multi-input Riccati equation to be solved, where all the inputs are coupled together. From the algorithm point of view, a relevant difference between the standard  $\mathcal{H}_2$  problem and the  $\mathcal{H}_\infty$  is related to the optimization process: due to the simultaneous minimization/maximization underlying the  $\mathcal{H}_\infty$ , we do not seek for a minimum but for a saddle point of the objective function (see for instance [33]). Finally, note that the both the disturbances  $\mathbf{B}_d$  and the  $\mathbf{B}_u$  can be multiple; also in this case, a centralized approach needs to be used.

Algorithm	IC	Grad.	Loops	Notes
DA	Random at each loop	$m \times n_t$	$n$	$\mathbf{K}$ is solution of a linear system.
ADA	$-\mathbf{R}_{i,i}^{-1} \mathbf{B}_i$	$m \times n_t$	$m$	Each row of $\mathbf{K}$ is solution of one iteration-loop.
MR	Random at each iter.	$m \times n$	1	$\mathbf{K}$ is solution of one iteration-loop, $\forall m$ .

**Table 1** The table summarizes the main features of the algorithms compared in Sec. 4.3: the column *IC* indicates the initial conditions used; the third column *Grad.* indicates the dimensions of the gradients used during the process of update; in the column *Loops*, the number of iteration-loops required for computing the control gains of dimension  $n \times m$  is indicated. In the last column, the characterizing features of the methods are highlighted.

## 4 Comparisons among algorithms

This section provides numerical results for assessing the performance of the ADA algorithm in presence of multiple inputs and centralized control. We consider a toy-problem and compare ADA with a recent algorithm proposed by [34]. The test-bed is mainly meant at verifying the speed of the convergence for the different algorithms.

### 4.1 Full-order controllers using stochastic gradients

The algorithm introduced by [34], and here indicated as MR, is used for comparison. The algorithm computes the control gain by updating at each step the solution as

$$\mathbf{K}^{i+1} = \mathbf{K}^i - \beta^i \left( \nabla_{\mathbf{K}} \tilde{\mathcal{J}} \right)^i, \quad (25)$$

where  $\nabla_{\mathbf{K}} \tilde{\mathcal{J}}$  is the gradient for the update at each step of the iteration. The basic formulation is proposed in appendix B. In principle, the algorithm suffers of the same limitations of the DA iteration: the identified solution depends by the initial condition. The problem is circumvented by choosing a new initial condition randomly at each iteration. The methods can be interpreted as a stochastic gradient descent method; at each step of the iteration, an initial state is chosen for which the gradient is computed. Using different initial states at every iteration, the final feedback matrix will be not depend from any of the initial conditions.

### 4.2 Implementation

The implemented routines are in prototypical form. The main idea is to analyze the scaling of the computational costs as a function of the number of actuators  $m$  and compare the resulting trends. Following this rational, a simple steepest

descent algorithm, where the step along the gradient direction is computed as

$$\beta = \frac{(d\mathcal{J}^i)^T d\mathcal{J}^i}{(d\mathcal{J}^i)^T d\mathcal{J}^{i+1}}. \quad (26)$$

The variable  $d\mathcal{J}^{i+1}$  indicates the gradient at the current step of the iteration. In principle, acceleration techniques might be applied (for instance, implicit restarting or Nesterov's accelerated methods) for further improving the performance of the algorithms. Two stop-criteria are chosen: the difference of the control cost between two iterations and the norm of the gradient. The iterations stop when one of the two criteria is fulfilled below a chosen tolerance.

In table 1, we highlight the main features of the algorithms. For all the cases the centralized/decentralized synthesis needs to be imposed when choosing the actuators of each single-input sub-system. In this section we only consider the centralized version of the algorithm that is the most expensive from the computational point of view. Note that for the MR algorithm the basic form is analysed, so without considering a pre-determined structure for the kernel; due to the random selection of the initial condition, the assessment of the performance is done by considering 10 simulations for each of the analysed cases.

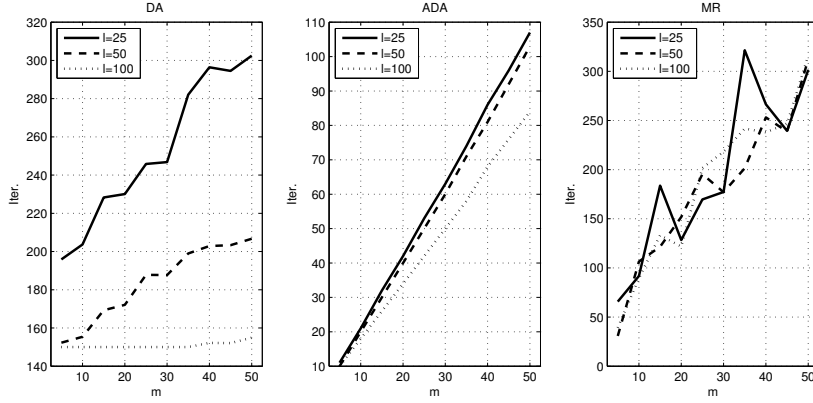
#### 4.3 Numerical example: a distributed system

A toy-problem mimicking a distributed system of dimension  $n = 50$  is considered. The system matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is in tri-diagonal form

$$\mathbf{A} = \begin{bmatrix} 1 & -4 & 0 & & & & \\ & -1 & -4 & 1 & & & \\ & & 1 & -4 & -2 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & -4 & -2 \\ & & & & & 8 & -8 & -1 \\ & & & & & & 0 & -2 & 1 \end{bmatrix}.$$

The matrix  $\mathbf{B} \in \mathbb{R}^{n \times m}$  contains non-null, unitary entries only along the main diagonal. The chosen system resembles the one used in [34]. The number of actuators is indicated by  $m$ ; 10 cases are analysed, where the number of actuators is chosen as  $m = [5, 10, 15, \dots, 50]$ . For the last case,  $n = m$ . The system is marched in time until the final time  $T = 20$ , with  $n_t = 501$ . The value of the final time was deemed sufficient for properly approximating the solution.

The solution of the optimal problem is computed for three different penalty kernels, namely  $l = [25, 50, 100]$  and benchmarked against the Riccati solution, obtained in Matlab using `care.m`.



**Fig. 2** Convergence of the iterative algorithms. The number of iterations as a function of the number of actuators  $m$  and control penalty  $l$  is considered for the algorithms described in Sec. 3.1 and Sec. 4.3.

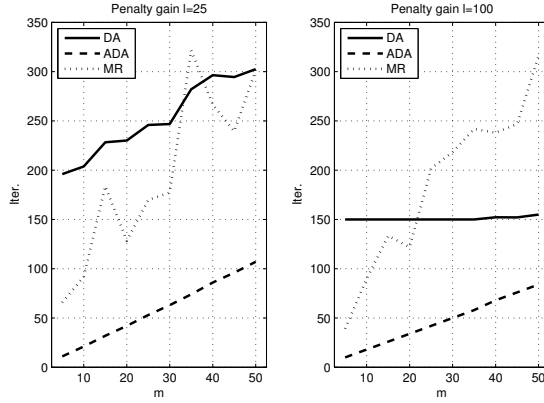
In Fig. 2, the number of iterations necessary for convergence is shown as a function of the number of actuators  $m$ . The chosen tolerance, given as the difference between the cost at  $i$  and  $i - 1$ , is  $\varepsilon = 1.0 \times 10^{-6}$ . All the algorithms scale with the number of actuators  $m$ . In particular, it is possible to observe a remarkable regularity for the ADA algorithm, that scales linearly with  $m$ ; different control penalties require different number of iteration. In particular, decreasing  $l$  requires a greater number of iterations. The behaviour was already observed in ADA, see [25]. This same behaviour is observed in the DA algorithm; in fact, only at  $l = 100$  is observed a constant number of iterations with  $m$ , while a larger number of iterations is required with the increasing number of actuators for the solution of the problem at lower  $l$ .

The DA algorithm requires on average more iterations when compared to the other iterative methods. This is not surprising, as  $n$  loops are required. However, the main bottleneck of DA is the final solution of a linear system, together with the storage of a large matrix containing along the columns the  $n$  initial conditions; in that sense, the use of the DA algorithm is unfeasible for larger system and is considered here only for the sake of completeness of our discussion.

The MR algorithm has a less clear behaviour, due to the choice of the initial conditions; an increase of the number of iterations is observed with  $m$  also for this case. The choice of the penalty does not affect the number of iterations required. This is clearer when observing Fig. 3, where the iteration number as a function of  $m$  is shown for two different penalty gains.

In conclusion, we can summarize the performance of three algorithms as follows

- For all the cases, an increase of the total number of iterations with  $m$  is observed. This is expected in ADA, where  $m$  iteration-loop are required,



**Fig. 3** Convergence of the iterative algorithms. The data shown in Fig. 2 are considered for  $l = 25$  and  $l = 100$ . The ADA algorithm is characterized by high scalability with the number of actuators and a number of iteration smaller with respect to the alternative choices.

while it is less obvious for the MR case, where only one iteration-loop is required regardless of  $m$ .

- ADA scales linearly with  $m$ . This is not the case for the MR algorithm, due to the stochastic gradient application.
- For the MR algorithm the averaged number of iterations is not affected by the control effort applied (i.e., the choice of the control penalty  $l$ ).

## 5 Numerical example: the two-dimensional Kuramoto-Sivashinsky

In this section, a modified version of the Kuramoto-Sivashinsky equation is used for testing the multi-input case, mimicking the three-dimensional setup sketched in Fig. 1; this proof-test provides the design feasibility in larger computational domains.

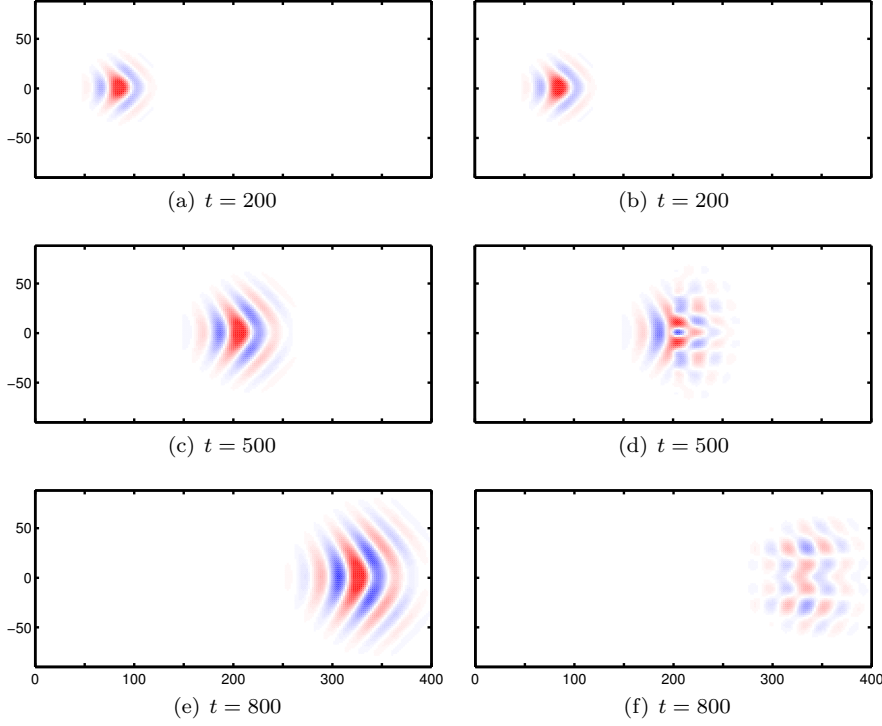
The flow field is approximated in a two-dimensional domain  $x$ - $z$  plane, and governed by the equation

$$\frac{\partial v}{\partial t} = -V \frac{\partial}{\partial x} \left( v - \frac{1}{8P} \frac{\partial^2 v}{\partial z^2} \right) - \frac{1}{R} \left( P \frac{\partial^2 v}{\partial x^2} + \frac{\partial^4 v}{\partial x^4} + S \frac{\partial^4 v}{\partial z^4} \right), \quad (27)$$

With respect to the original equation, the dynamics is linearized around the convective velocity,  $V = 0.4$ . The non-dimensional terms  $R$  and  $P$  are defined as

$$R = \frac{VL^3}{\mu} \quad P = \frac{\eta L^2}{\mu}, \quad (28)$$

with  $\eta$  representing an energy production term and  $\mu$  the dissipation;  $L$  is the reference length of the system. The parameters can be chosen ad-hoc such that the dispersion relation of the system mimics the evolution of travelling packets of Tollmien-Schlichting (TS) waves. Introducing the maximum streamwise



**Fig. 4** The propagation of a wavepacket in a flow governed by the modified Kuramoto-Sivashinsky equation is shown for a clean configuration (insets *a-c-e*) and the controlled case (insets *b-d-f*). The non-dimensional parameters in Eq. 28 and Eq. 29 are chosen for mimicking the propagation of Tollmien-Schlichting waves at  $Re = 1000$ , with  $\alpha_{max} = 1.68 \times 10^{-1}$ ,  $\beta_{max} = 2.15 \times 10^{-1}$  and  $\omega_{max} = 2.67 \times 10^{-3}$ .

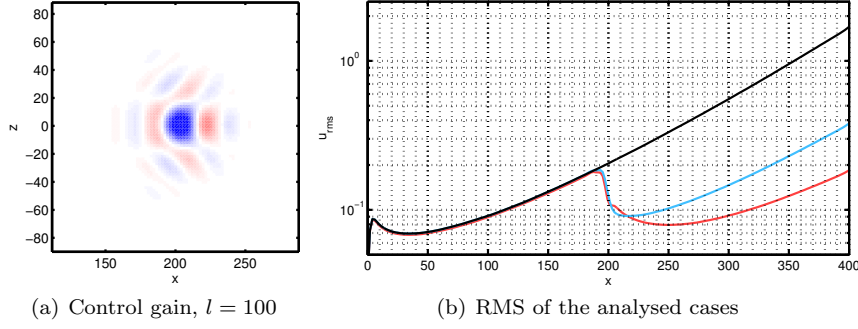
and spanwise wavenumbers as  $\alpha_{max}, \beta_{max} \in \mathbb{R}$  respectively, and the complex temporal frequency  $\omega$ , the non-dimensional numbers can be expressed as

$$R = \frac{P^2}{4\omega_{max}}, \quad P = \frac{2}{\alpha_{max}^2}, \quad S = \frac{\omega_{max}R}{\beta_{max}^4}, \quad (29)$$

with  $S$  providing the modulation along the spanwise direction of the wave. The numbers are chosen such that the dispersion relation closely resembles the behaviour in the horizontal plane of a TS wave evolving on a flat-plate at  $Re_{\delta^*} = 1000$  (see [6], [7]). More details are reported by [35].

The computational box extends along the streamwise direction in the interval  $x \in [0, 500]$  and  $z \in [-90, 90]$  along the spanwise direction; a grid with  $N_x = 256$  and  $N_z = 96$  is chosen. The spatial discretization is performed by means of a pseudo-spectral method, that includes a fringe region extending between  $x = 400$  and  $x = 500$ ; periodic boundary conditions are imposed along the spanwise direction. The time marching is performed using a three-steps





**Fig. 5** *Left*: the full-order control gain related to the actuator  $\mathbf{B}_u$  placed at  $\mathbf{x} = (200, 0)$  using the ADA, with a control penalty  $l = 100$ . *Right*: the root mean squares (RMS) of the velocity is shown as a function of the streamwise direction  $x$ . The value is computed using  $N = 10000$  snapshots. The clean configuration (black, solid line) is compared to the controlled cases designed using  $l = 100$  (red-solid line) and  $l = 500$  (blue-solid line).

Runge-Kutta scheme; the basic implementation can be found in the repository <https://github.com/nfabbiane/ks2D>.

### 5.1 A multi-input full-order controller

The resulting dynamics is depicted in Fig. 4*a-c-e*, where the impulse response of the system is shown at three different instants. The disturbance is introduced at  $\mathbf{x}_0 = (x_0, z_0) = (2.5, 0)$ , and its spatial distribution is modelled as

$$\mathbf{B} = \exp\left(-\frac{(x - x_0)^2}{\sigma^2} - \frac{(z - z_0)^2}{\sigma^2}\right), \quad (30)$$

with  $\sigma = 4$ . The dynamics of the wavepacket mimics the evolution of a TS wave, growing as it propagates downstream along the streamwise direction while extending along the spanwise direction with a backward bending. The corresponding energy growth associated with the evolving wavepacket is shown in Fig. 5*b* (black, solid line); the root-mean-square energy is obtained from the statistics associated with a stochastically driven simulation, averaged over a span of  $T = 10000$  time units.

A LQR controller is designed using the ADA algorithm. The controller is centralized: all the gains are designed such that the set of sensors and actuators is coupled. The setup closely resembles the one analysed by [6], [7] and reproduced schematically in Fig. 1: a row composed by  $n = 9$  localized actuators ( $\mathbf{B}_u$ ) is placed along the spanwise direction at  $x = 200$ , equi-spaced of  $\Delta z = 20$ . The same setup along the spanwise direction is chosen for the sensors  $\mathbf{C}_z$ , placed at  $x = 300$ . All the chosen elements are modelled as Gaussian distributions, Eq. 30. Nine iterations are set, one for each control gain. An example is given in Fig. 5*a*, where the control gain placed at  $\mathbf{x} = (200, 0)$ , with  $l = 100$ ; the controller shows the typical signature of the adjoint solution,

as it is bent backward with respect to the propagation of the direct solution. The performances of two controllers are shown in Fig. 5b, where the clean configuration is compared with two cases:  $l = 100$  (red solid line), and  $l = 500$  (blue solid line). The results are in qualitative agreement with [6]. More interestingly, it shows that the technique enables the computation of multi-input, full order controllers without using a preliminary model reduction step.

## 6 Conclusions

We extended the framework of the *Adjoint of the adjoint* algorithm to multivariate large systems, by highlighting how decentralized, centralized and coupled controllers can be computed for full-order, optimal control. These observations apply straightforwardly to the dual problem, the estimation problem, and to robust controllers belonging to the  $\mathcal{H}_\infty$  framework. The main advantage of the ADA algorithm (and its extensions) is the independence of the final solution from the initial conditions. From the physical point of view, this implies that it is not required any knowledge of the disturbances active in the flow. This feature makes the algorithm particularly appealing for the pre-assessment of the optimal performance of a controller, based on linear (or linearized) plant. Moreover, one can assess the controllability of the dynamical system by such techniques and estimate the efficiency of the resulting controller.

Two test cases are considered. A toy-problem mimicking a distributed system is analysed for assessing the convergence performance of ADA against an analogous algorithm by Mårtensson *et al.* [34], indicated as MR. The results show a remarkable scalability of ADA, with respect the number of actuators; with respect to the MR algorithm, the number of iterations required is on average three times smaller and - more importantly - it is not affected by the stochastic approach required by the MR algorithm for identifying the global solution. The second example is based on a modified version of the Kuramoto-Sivashinsky. This example reproduces a control setup based on multiple, localized sensors and actuators placed along the cross-flow direction. The feasibility of the approach is demonstrated suggesting that the technique might applied also to full three-dimensional cases in more complex fluid mechanics settings.

Future work will be devoted to the application of these methods in combination with sparsity promoting algorithms (see [36]). In our application, the coupling among the actuators is pre-determined and not optimized. However, it is possible to introduce the sensor/actuator pairing within the optimization process (pairing problem). Moreover, sensor and actuator placement for large-scale systems might be analysed starting from this application ([37], N. Fabiane, *private communication*, 2016).

**Acknowledgments** Anders Rantzer and his collaborators (Lund University) are acknowledged for interesting discussions and for bringing up their work

on optimal control synthesis in large systems. Nicolò Fabbiane (DAFE, On-  
era) provided the basic implementation of the two-dimensional Kuramoto-  
Sivashinsky equation. The first author thanks the support by the Agence Na-  
tionale de la Recherche (ANR) under the *CoolJazz* project, grant number  
ANR-12-BS09-0024.

## A Derivation of ADA algorithm using integration by parts

In this section we briefly summarize the original derivation of the ADA algorithm, as first  
proposed by [1]. We start from the direct-adjoint system

$$\frac{d\mathbf{q}}{dt} = \mathbf{A}\mathbf{q} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^H\mathbf{p}, \quad \text{with } \mathbf{q}(0) = \mathbf{q}_0, \quad (31a)$$

$$\frac{d\mathbf{p}}{dt} = -\mathbf{A}^H\mathbf{p} - \mathbf{Q}\mathbf{q}, \quad \text{with } \mathbf{p}(T) = 0, \quad (31b)$$

where the optimality condition has already been included. By introducing two variables,  
namely the state vectors  $\mathbf{a}(t)$  and  $\mathbf{b}(t)$ , we can recombine the system as

$$\int_0^T \mathbf{a}^H \left( \frac{d\mathbf{q}}{dt} - \mathbf{A}\mathbf{q} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^H\mathbf{p} \right) dt + \int_0^T \mathbf{b}^H \left( \frac{d\mathbf{p}}{dt} + \mathbf{A}^H\mathbf{p} + \mathbf{Q}\mathbf{q} \right) dt.$$

Integration by parts allows rewriting the relation as

$$\int_0^T \mathbf{q}^H \left( -\frac{d\mathbf{a}}{dt} - \mathbf{A}^H\mathbf{a} + \mathbf{Q}\mathbf{b} \right) dt + \int_0^T \mathbf{p}^H \left( -\frac{d\mathbf{b}}{dt} + \mathbf{A}\mathbf{b} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^H\mathbf{a} \right) dt + \\ (\mathbf{a}^H\mathbf{q} + \mathbf{b}^H\mathbf{p})_T - (\mathbf{a}^H\mathbf{q} + \mathbf{b}^H\mathbf{p})_0$$

Gathering the resulting state equations for  $\mathbf{a}(t)$  and  $\mathbf{b}(t)$ , and changing the variables  $\mathbf{a} \rightarrow \tilde{\mathbf{p}}$ ,  
 $\mathbf{b} \rightarrow -\tilde{\mathbf{q}}$ , allow us to write the following system

$$\frac{d\tilde{\mathbf{q}}}{dt} = \mathbf{A}\tilde{\mathbf{q}} + \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^H\tilde{\mathbf{p}}, \quad \tilde{\mathbf{q}}(0) = \tilde{\mathbf{q}}_0, \quad (32a)$$

$$-\frac{d\tilde{\mathbf{p}}}{dt} = \mathbf{A}^H\tilde{\mathbf{p}} + \mathbf{Q}\tilde{\mathbf{q}}, \quad \tilde{\mathbf{p}}(T) = 0, \quad (32b)$$

$$\tilde{\mathbf{p}}(T)^H\mathbf{q}(T) - \tilde{\mathbf{q}}(T)^H\mathbf{p}(T) = \tilde{\mathbf{p}}(0)^H\mathbf{q}(0) - \tilde{\mathbf{q}}(0)^H\mathbf{p}(0). \quad (32c)$$

Equations 32 correspond to the ones of the original optimal control problem 31. The last  
relation 32(c) is exactly the condition recovered by symplectic product in Eq. (11); thus,  
the observations done in Eqs. (11–12) are valid also for this case.

## B Full-order controllers using stochastic gradients

In this appendix, the algorithm discussed by Mårtensson and Rantzer in [34] is derived  
following the Lagrangian approach. For sake of conciseness, we will indicate the algorithm  
with the acronym MR. The state equation is re-written as

$$\frac{d\mathbf{q}}{dt} = (\mathbf{A} + \mathbf{B}\mathbf{K})\mathbf{q} \quad \text{with } \mathbf{q}(0) = \mathbf{q}_0, \quad (33)$$

The problem is rewritten now assuming the control kernel  $\mathbf{K}$  as unknown of the problem,  
replacing  $\mathbf{u}(t)$ . The cost function is now

$$\mathcal{J} = \frac{1}{2} \int_0^T \left( \mathbf{q}^H \mathbf{Q} \mathbf{q} + \mathbf{q}^H \mathbf{K}^H \mathbf{R} \mathbf{K} \mathbf{q} \right) dt = \frac{1}{2} \int_0^T \mathbf{q}^H \left( \mathbf{Q} + \mathbf{K}^H \mathbf{R} \mathbf{K} \right) \mathbf{q} dt, \quad (34)$$

and the resulting augmented Lagrangian is

$$\tilde{\mathcal{J}} = \mathcal{J} - \int_0^T \mathbf{p}^H \left( \frac{d\mathbf{q}}{dt} - \mathbf{A}\mathbf{q} - \mathbf{B}\mathbf{K}\mathbf{q} \right) dt.$$

The resulting system is obtained by considering the gradients with respect to  $\mathbf{p}$ ,  $\mathbf{q}$  and  $\mathbf{K}$

$$\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{p}} = \frac{d\mathbf{q}}{dt} - \mathbf{A}\mathbf{q} - \mathbf{B}\mathbf{K}\mathbf{q}, \quad (35a)$$

$$\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{q}} = \frac{d\mathbf{p}}{dt} + \mathbf{A}^H \mathbf{p} + \mathbf{K}^H \mathbf{B}^H \mathbf{p} + (\mathbf{Q} + \mathbf{K}^H \mathbf{R} \mathbf{K}) \mathbf{q}, \quad (35b)$$

$$\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{K}} = \int_0^T (\mathbf{R}\mathbf{K}\mathbf{q} + \mathbf{B}^H \mathbf{p}) \mathbf{q}^H dt. \quad (35c)$$

Zeroing the gradients, the system can be arranged in matrix form as

$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{pmatrix} = \begin{bmatrix} \mathbf{A} + \mathbf{B}\mathbf{K} & \mathbf{0} \\ -(\mathbf{Q} + \mathbf{K}^H \mathbf{R} \mathbf{K}) & -(\mathbf{A} + \mathbf{B}\mathbf{K})^H \end{bmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix}.$$

The solution is updated by using 35c as

$$\mathbf{K}^{i+1} = \mathbf{K}^i - \beta \nabla_{\mathbf{K}} \tilde{\mathcal{J}}. \quad (36)$$

The iteration is usually initialized with a guess for  $\mathbf{K}$ , unless the system under consideration is asymptotically stable; in the latter case, the controller is designed for modifying the transient dynamics and the starting guess is simply a null vector. In this formulation, the problem depends on the initial conditions. The solution is not guaranteed to be the one corresponding to the global minimum for a given initial condition. This drawback can be circumvented by using *ad-hoc* strategies as the stochastic gradient approach described in [34]. Moreover, with respect to the standard algorithm, the MR algorithm does not require the solution of Lyapunov equations, replaced with an approximation based on the adjoint and direct solutions.

### B.1 Relation with the ADA algorithm: generalizing the sensitivity approach

The symplectic product adopted in ADA can be introduced also for the system in Eq. B. It is possible to show that the relation

$$\tilde{\mathbf{p}}(t)^H \mathbf{q}(t) = \tilde{\mathbf{q}}(t)^H \mathbf{p}(t) \quad \forall t, \quad (37)$$

is fulfilled also in this case. Considering again the gradient,

$$\nabla_{\mathbf{K}} \tilde{\mathcal{J}} = \int_0^T (\mathbf{R}\mathbf{K}\mathbf{q} + \mathbf{B}^H \mathbf{p}) \mathbf{q}^H dt, \quad (38)$$

it is null  $\forall \mathbf{q}$  if

$$\mathbf{R}\mathbf{K}\mathbf{q}(t) = -\mathbf{B}^H \mathbf{p}(t). \quad (39)$$

By rescaling the relation via  $\mathbf{R}^{-1}$ , we recover the equivalence between the product and the optimality condition at  $t = 0$

$$\tilde{\mathbf{p}}(0)^H \mathbf{q}_0 = \tilde{\mathbf{q}}(0)^H \mathbf{p}_0, \quad (40a)$$

$$\mathbf{K}\mathbf{q}_0 = -\mathbf{R}^{-1} \mathbf{B}^H \mathbf{p}_0. \quad (40b)$$

Thus, also for the basic equation of the MR algorithm, introducing as initial conditions of the system  $\tilde{\mathbf{q}} = -\mathbf{R}_{i,i}^{-1} \mathbf{B}_i$ , the iteration enables to compute the  $i$ -th kernel  $\mathbf{K}$  as solution

of the adjoint equation at the final time  $T$ . All the observations done for the multi-input, coupled version of the ADA algorithm are valid also for this algorithm. Interestingly, this algorithm produces the control kernels in two different ways: i) as adjoint solution; ii) as result of the gradient-based iteration

$$\mathbf{K}^{n+1} = \mathbf{K}^n - \beta \nabla_{\mathbf{K}} \mathcal{J}. \quad (41)$$

The sensitivity with respect to the initial conditions applied to the optimal control problem leads to the same results with different choice of gradients. The gradient with respect to  $\mathbf{K}$  can be rewritten, using the chain-rule, as

$$\frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{K}} = \frac{\partial \tilde{\mathcal{J}}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{K}} = \left( \mathbf{R} \mathbf{K} \mathbf{q} + \mathbf{B}^H \mathbf{p} \right) \mathbf{q}^H. \quad (42)$$

The underlying system of equations is equivalent in the two cases.

## References

1. T. Bewley, P. Luchini, J. Pralits, Methods for solution of large optimal control problems that bypass open-loop model reduction, *Meccanica* 51 (12) (2016) 2997–3014.
2. F. Abergel, R. Temam, On some control problems in fluid mechanics, *Theor. Comp. Fluid Dyn.* 1 (1990) 303–325.
3. J. Kim, T. R. Bewley, A Linear Systems Approach to Flow Control, *Ann. Rev. Fluid Mech.* 39 (2007) 39–383. doi:10.1146/annurev.fluid.39.050905.110153.
4. N. Fabbiane, O. Semeraro, S. Bagheri, D. S. Henningson, Adaptive and model-based control theory applied to convectively unstable flows, *Applied Mechanics Reviews* 66 (6) (2014) 060801.
5. D. Sipp, P. J. Schmid, Linear closed-loop control of fluid instabilities and noise-induced perturbations: A review of approaches and tools, *Applied Mechanics Reviews* 68 (2) (2016) 020801.
6. O. Semeraro, S. Bagheri, L. Brandt, D. S. Henningson, Feedback control of three-dimensional optimal disturbances using reduced-order models, *Journal of Fluid Mechanics* 677 (2011) 63–102.
7. O. Semeraro, S. Bagheri, L. Brandt, D. S. Henningson, Transition delay in a boundary layer flow using active control, *Journal of Fluid Mechanics* 731 (2013) 288–311.
8. F. L. Lewis, L. V. Syrmos, *Optimal Control*, John Wiley & Sons, New York, 1995.
9. P. Benner, J. Li, T. Penzl, Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems, *Numer. Linear Algebra Appl.* 15 (2008) 755–777.
10. P. Benner, Solving large-scale control problems, *Control Systems IEEE* 24 (1) (2004) 44–59.
11. H. T. Banks, K. Ito, A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems, *SIAM J. Control and Optimization* 29 (3) (1991) 499–515.
12. I. Akhtar, J. Borggaard, J. A. Burns, H. Imtiaz, L. Zietsman, Using functional gains for effective sensor location in flow control: a reduced-order modelling approach, *Journal of Fluid Mechanics* 781 (2015) 622–656.
13. M. Högberg, T. R. Bewley, D. S. Henningson, Linear feedback control and estimation of transition in plane channel flow, *J. Fluid Mech.* 481 (2003) 149–175.
14. B. Bamieh, F. Paganini, M. Dahleh, Distributed control of spatially invariant systems, *Automatic Control, IEEE Transactions on* 47 (7) (2002) 1091–1107. doi:10.1109/TAC.2002.800646.
15. M. Chevalier, J. Höpfner, E. Åkervik, D. S. Henningson, Linear feedback control and estimation applied to instabilities in spatially developing boundary layers, *J. Fluid Mech.* 588 (2007) 163–187.
16. A. Monokrousos, L. Brandt, P. Schlatter, D. S. Henningson, Dns and les of estimation and control of transition in boundary layers subject to free-stream turbulence, *International Journal of Heat and Fluid Flow* 29 (3) (2008) 841–855.

17. T. Bewley, J. Pralits, P. Luchini, Minimal-energy control feedback for stabilization of bluff-body wakes based on unstable open-loop eigenvalues and left eigenvectors, in: *Proceedings of the Fifth Conference on Bluff Body Wakes and Vortex-Induced Vibrations (BBVIV5)*, 2007, pp. 129–132.
18. M. Carini, J. Pralits, P. Luchini, Feedback control of vortex shedding using a full-order optimal compensator, *Journal of Fluids and Structures* 53 (2015) 15–25.
19. M. Carini, C. Airiau, A. Debien, L. O., P. J. O., Global stability and control of the confined turbulent flow past a thick flat plate, *Accepted, Physics of Fluids*.
20. P. Luchini, A. Bottaro, Adjoint equations in stability analysis, *Annual Review of fluid mechanics* 46 (1) (2014) 493.
21. T. R. Bewley, P. Moin, R. Temam, Dns-based predictive control of turbulence: an optimal benchmark for feedback algorithms, *Journal of Fluid Mechanics* 447 (2) (2001) 179–225.
22. S. Cherubini, J.-C. Robinet, P. De Palma, Nonlinear control of unsteady finite-amplitude perturbations in the blasius boundary-layer flow, *Journal of Fluid Mechanics* 737 (2013) 440–465.
23. J. Kim, D. J. Bodony, J. B. Freund, Adjoint-based control of loud events in a turbulent jet, *Journal of Fluid Mechanics* 741 (2014) 28–59.
24. J. O. Pralits, P. Luchini, Riccati-less optimal control of bluff-body wakes, in: *Seventh IUTAM Symposium on Laminar-Turbulent Transition*, Springer Netherlands, 2010, pp. 325–330.
25. O. Semeraro, J. O. Pralits, C. W. Rowley, D. S. Henningson, Riccati-less approach for optimal control and estimation: an application to two-dimensional boundary layers, *Journal of Fluid Mechanics* 731 (2013) 394–417.
26. T. Glad, L. Ljung, *Control theory*, CRC press, 2000.
27. K. Zhou, J. C. Doyle, K. Glover, *Robust and Optimal Control*, Prentice Hall, New Jersey, 2002.
28. K. Mårtensson, *Gradient methods for large-scale and distributed linear quadratic control*, Ph.D. thesis, Lund University (2012).
29. W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, 3rd Edition, Cambridge University Press, 2007.
30. S. Skogestad, I. Postlethwaite, *Multivariable Feedback Control, Analysis to Design*, 2nd Edition, Wiley, 2005.
31. S. Bagheri, L. Brandt, D. S. Henningson, Input–output analysis, model reduction and control of the flat-plate boundary layer, *Journal of Fluid Mechanics* 620 (2009) 263–298.
32. S. Bagheri, J. Hoepffner, P. J. Schmid, D. S. Henningson, Input-Output analysis and control design applied to a linear model of spatially developing flows, *Appl. Mech. Rev.* 62. doi:10.1115/1.3077635.
33. S. Gumusoy, D. Henrion, M. Millstone, M. L. Overton, Multiobjective robust control with hifoo 2.0, *IFAC Proceedings Volumes* 42 (6) (2009) 144–149.
34. K. Mårtensson, A. Rantzer, A scalable method for continuous-time distributed control synthesis, in: *2012 American Control Conference (ACC)*, IEEE, 2012, pp. 6308–6313.
35. J. Sundin, Controlling the laminar-to-turbulent transition in a fluid flow, *Tech. rep.*, KTH.
36. F. Lin, M. Fardad, M. R. Jovanovic, Design of optimal sparse feedback gains via the alternating direction method of multipliers, *Automatic Control, IEEE Transactions on* 58 (9) (2013) 2426–2431.
37. K. K. Chen, C. W. Rowley, H 2 optimal actuator and sensor placement in the linearised complex ginzburg-landau system, *Journal of Fluid Mechanics* 681 (2011) 241–260.